

On Combinatorial Auction and Lagrangean Relaxation for Distributed Resource Scheduling

Erhan Kutanoglu and S. David Wu

Manufacturing Logistics Institute
Department of Industrial and Manufacturing Systems
Engineering
Lehigh University

Abstract

Most existing methods for scheduling are based on centralized or hierarchical decision making using monolithic models. In this study, we investigate a new method based on a distributed and locally autonomous decision structure using the notion of *combinatorial auction*. In *combinatorial auction* the bidders demand a combination of dependent objects with a single bid. We show that not only can we use this auction mechanism to handle complex resource scheduling problems, but there exist strong links between combinatorial auction and Lagrangean-based decomposition. Exploring some of these properties, we characterize combinatorial auction using auction protocols and payment functions. This study is a first step toward developing a distributed scheduling framework that maintains system-wide performance while accommodating local preferences and objectives. We provide some insights to this framework by demonstrating four different versions of the auction mechanism using job shop scheduling problems.

Key words: Auction theory, mechanism design, job shop scheduling, distributed scheduling, Lagrangean relaxation, subgradient optimization.

1 Introduction

Distributed decision making has attracted much attention in recent years due to the rapid advancement of computing and communication technologies. Local decision makers responsible for a segment of the decision process now have direct access to powerful decision tools and workstations. The idea of making decisions in a heterogeneous, distributed fashion becomes both desirable and practical. Under such systems, decision makers may contemplate and incorporate their local set of constraints, preferences, and objectives into the decision process. Nonetheless, local considerations are but loosely linked to global company interests, and decision makers often have conflicting interests over their shared resources. Two important research issues are thus, aligning global and local interests, and resolving peer-to-peer conflicts in resource sharing.

In this paper, we will focus our attention on resource scheduling problems. Resource scheduling is a critical component of manufacturing information systems. The structure of a scheduling system affects directly how production functions are planned and carried out. Most existing methods for resource scheduling take one of the following two approaches: (1) centralized or hierarchical decision making based on some monolithic optimization model, or (2) distributed but ad hoc decision making designed to fit the existing organizational structure. Monolithic models are often found in the academic literature especially in Operations Research, while distributed or localized methods can be frequently observed in industry. We are interested in mechanisms that allow resource scheduling to be locally autonomous, and at the same time aligned with global interests.

The need for distributed resource scheduling is most evident in manufacturing systems with complex product/supply structures, and where customer service is of high priority. Electronics and automotive manufacturing fall into this category. In these operations, firms often have designated product managers each responsible for a set of similar product types. By design, each product manager is motivated to satisfy his/her own customers' requirements. Each product manager must deal with distinctively different set of issues due to different market conditions, lead time pressure, and processing requirements. It is seldom economical to construct dedicated production lines for each product family using duplicated resources. Consequently product managers must share and compete for production resources in a regular basis. To satisfy the conflicting objectives of overall resource efficiency and individual product performance (e.g., market responsiveness, cus-

tomers satisfaction), a new generation of scheduling system is in desperate need.

In this paper, we study a distributed scheduling framework, which we believe is more suitable for the manufacturing decision structure. The basic idea is to localize and distribute the functionality of operational scheduling decisions, leaving the complexity to local decision makers, while maintaining a simple and generic coordination mechanism at a central site. The proposed method is distributed in the sense that each local decision maker bases his/her decisions on a local utility which is defined based on both local preferences and global constraints. Specifically, each decision-maker has a local problem to maximize his/her expected total reward subject to local constraints. This is then communicated to the coordination mechanism as a “bid”. The coordination mechanism or an “auctioneer” is a bid processor that makes resource allocation based on an iterative auction process using the bidding information. We will demonstrate that traditional optimization models for scheduling can be viewed as special cases of the proposed auction structure, and well-known optimization techniques such as Lagrangean relaxation can be adapted to handle iterative auction.

2 A Survey of Related Literature

The notion of decentralized decision making can be found in early OR literature as well as in much recent work in Economics and Computer Science. To put our proposed method in the context of existing literature, we will provide a survey of the latter two areas: agent-based scheduling emerged from the Computer Science community, and auction theory and equilibrium from Economics and Game Theory.

2.1 Distributed and Agent-based Scheduling

Distributed scheduling has attracted much attention recently in the computer science community. Methods based on *multi-agent systems* have emerged in the area of Distributed Artificial Intelligence (DAI). The idea is to decouple the scheduling task into subproblems each solved by a local agent. These approaches are rooted from the constraint satisfaction version of the scheduling problem [49, 26, 27]. The optimization version of the problem has

rarely been attempted in DAI [49, 28]. Sycara, et al. [49] revise the micro-opportunistic scheduling approach studied in [44] to develop a distributed and multi-agent scheduling system. Liu and Sycara [28] propose a multi-agent approach for specially structured bottleneck job shop scheduling problem with weighted tardiness objective.

Another line of research, also rooted from DAI, is *market-based* distributed problem solving. In this approach the communication among the problem-solving agents is governed by a computational economic process. The *contract net* approach proposed by Davis and Smith [10] is among the pioneer work in this area. The contract net model has been proposed for *dynamic on-line* manufacturing scheduling where an agent responsible for a manufacturing step issues a request-for-bids to all agents who can supply the input. The best bidder is then selected based on a certain criteria [36, 37, 1]. Upton, Barash and Matheson [50] study a market mechanism for parallel machine scheduling. They propose a particular bid calculation based on the earliest expected completion time. A similar auction-based approach for setup time determination is proposed by Wang and Veeramani [52].

Neiman, et al. [34] investigate a multi-agent distributed scheduling system where each agent is responsible for a set of resources. To perform its assigned tasks, an agent negotiates with other agents for its required resources. Sandholm [45] proposes a similar dynamic scheduling approach for the vehicle routing problem.

Another line of research in DAI uses the notion of economic equilibrium [54] to model and analyze the interactions among distributed agents. Wellman [53] proposes such a approach which finds equilibrium prices for multicommodity flow problems. More recently, Ygge and Akkermans [56] propose a resource-oriented equilibrium-seeking approach for power load management problems. This is a quantity tatonnement approach for Walrasian or competitive equilibrium. In a recent study, Wellman, et al. [55] develop price equilibrium for single-resource preemptive scheduling problems.

2.2 Auction Mechanism Design and Auction Algorithms

McAfee and McMillan [32] define auction as a market institution with an explicit set of rules determining resource allocation and prices on the basis of bids from market partici-

pants. The literature on auction theory flourished after the seminal paper by Vickrey [51], in which he introduces the concept of *second-price* auctions. A bibliography by Stark and Rothkopf [48] lists nearly 500 papers over the next two decades after Vickrey's study. In another survey, Engelbrecht-Wiggans [12] reviews main theoretical results and research efforts on auctions and bidding. Smith [47] discusses auction research in experimental economics published between 1962 and 1990. Most studies in this area focus on single-object simple auctions [33, 13]. A more recent and broader scoped review on competitive bidding can be found in [41].

If we consider scheduling problems as assigning time slots on unit-capacity machines (objects) to a set of competing jobs (bidders), we can see that it is at least as complex as *multi-object auctions* which deal with independent-valued, indivisible objects. Hylland and Zeckhauser [20] consider the problem of allocating individuals to positions with limited capacities. They propose an auction-assignment mechanism where the individuals respond to the system with true valuations. Leonard [25] investigates the problem of eliciting true preferences for an assignment problem in a similar setting. He presents a generalization of Vickrey's second-price auction mechanism. Demange, Gale and Sotomayor [11] propose a *dynamic* or *progressive auction* for the multi-item case to achieve incentive-compatible minimum price equilibrium. Instead of a single-pass sealed bid auction as in Leonard [25], they present two progressive auction mechanisms, one exact and another approximate. Sankaran [46] proposes an improvement on the exact auction mechanism based on a labeling algorithm.

The equivalence of multi-object auctions and the classical assignment problem led Bertsekas to develop so called *auction algorithm* [3]. Auction algorithm is similar to the approximate auction mechanism proposed by Demange, Gale and Sotomayor [11]. Bertsekas show that the price vector obtained at the end of the auction is the approximate optimal dual vector of the primal assignment problem. This, in fact, is a restatement of the result of Demange, Gale and Sotomayor. In this context, we can say that finding the equilibrium prices in an optimal auction design problem is equivalent to finding the dual optimal solution for the assignment problem. Bertsekas later extends auction algorithm for other network flow problems such as shortest path, transportation and minimum cost flow [4, 5, 6].

In general scheduling problems a job agent may demand a combination of time slots to process its operations. Consequently we must deal with auctions in which agents bid for

multiple items that have inter-dependent valuations. This motivates the investigation of *combinatorial auction* in which the bidders demand a set or a combination of indivisible objects with a single bid. Banks, Ledyard and Porter [2] consider allocating multiple resources each with *divisible* capacity when there are uncertainties in supply or demand. They generalize Vickrey's second-price auction for this environment where each bidder demands a combination of some portions of each resource with a single bid. Rothkopf, Pekec and Harstad [42] investigate simultaneous combinatorial auctions in which bidders submit bids for a combination of indivisible objects. They show that finding the revenue-maximizing set of nonconflicting bids is NP-Hard. They examine different bid structures to find computationally tractable auction types. Applications of auction mechanisms to optimization has also been proposed by Graves, Schrage, and Sankaran [16] for class scheduling, and Rassenti, Smith, and Bulfin [40] for the allocation of airport time slots to airlines.

In a majority of these problems, competitive (Walrasian) price equilibria are known to exist and auction procedures aim to reach one of these equilibria operate in an efficient manner. In this respect, price-directed auctions can be viewed as tâtonnements designed to reach market clearing (equilibrium) prices in an exchange economy [31]. Several researchers study the correspondence between equilibrium conditions and the convergence properties of auction. Most of these work are done in the context of exchange economy with interdependent and indivisible objects (c.f., [7, 29, 30, 18, 19]). These studies show that a price equilibrium may not exist in general combinatorial auctions where agents demand a bundle of interdependent items. Gul and Stacchetti [19] show that in such an economy unless the agents' utility functions satisfy conditions called *gross substitutes* or *no complementarities* the existence of a price equilibrium is not guaranteed. Bikhchandani and Mamer [7] show that market clearing prices exist *if and only if* the solution value of a centralized integer program coincide with that of its LP relaxation. This condition significantly lower the expectation of reaching price equilibrium using any simple auction mechanism.

3 Problem Statement and Solution Methodology

3.1 Problem Statement

We consider the classical job shop scheduling problem (JSP) where a set of jobs is to be completed and each job requires a set of machines for a certain period of time for processing. Each *job* consists of a series of operations that represent the production steps of the job. Each *operation* needs a certain *machine* for a certain time period called *processing time*. Once an operation is started, no to interruptions are allowed until completion, i.e., no preemption. Job processing times and their corresponding machine requirements represent the job *routing*. The prescribed sequence of operations in each job defines the *precedence constraints*. Each machine has a capacity of one, i.e., it can process one and only one operation at a time. The scheduling objective is to minimize total weighted tardiness. When all jobs have identical routing, JSP is reduced to a flow shop scheduling problem. If each job requires only a single stage operation on one particular machine, or a dominant resource forms the bottleneck, then JSP is reduced to a single machine problem. For a complete taxonomy of production scheduling problems, see Pinedo [38].

3.2 A Combinatorial Auction Mechanism for Scheduling

We propose a general auction mechanism for JSP using the notion of multi-item combinatorial auction. The proposed mechanism is progressive or dynamic [11] since it involves some number of iterations before allocating objects to bidders. We define “objects” as discrete time slots on the machines, and “bidders” as individual jobs. An auctioneer is the coordinating agent or the seller who iteratively updates the prices of objects starting from initial reservation prices. Based on current pricing, each job tries to find the best combination of time slots on the machines (objects) so as to maximize its own utility function. The auctioneer evaluates bids from all jobs and updates the reservation prices according to the conflicts among their requests. This process repeats in an iterative fashion until it finds a conflict-free allocation [11]. Obviously, objects have inter-dependent values, and different combination of objects presents different values for the jobs as is the case in combinatorial auctions [40, 42]. The precedence and non-preemption constraints restrict the combination of time slots that each job can bid

Table 1: Notation used throughout the paper

i	Job (bidder) index, $i = 1, \dots, N$ where N is number of jobs
j	Operation index, $j = 1, \dots, n_i$ where n_i is number of operations of job i
t	Time slot index, $t = 1, \dots, T$ where T represents the length of the planning horizon during which all the jobs can be completed (i.e. some reasonable upper bound on the makespan of the problem).
k	Machine index, $k = 1, \dots, M$ where M is number of machines (Hence there are TM time slots for bidding).
W_i	Weight or tardiness penalty of job i
d_i	Due date of job i
o_{ik}	The operation of job i which requires machine k
m_{ij}	Machine required for operation j of job i , i.e. $o_{ik} = j$ if $m_{ij} = k$
p_{ij}	Processing time of operation j of job i
$B_{ij;a,b}$	Operation bid, i.e. a combination of time slots from time slot a to time slot b for operation j of job i
B_i	Job bid, i.e. collection of operation bids (A combination of time slots demanded by job i)

In the following, we state more formally the price directed auction procedure for the weighted tardiness job shop scheduling problem. We adopt the notation shown in Table 1. The set of objects (time slots available from the machines) can be defined as a set of pairs (machine, time slot). Hence each possible bid B_i from job i is a subset of the following object set:

$$O = \{(k, t) : 1 \leq k \leq M, 1 \leq t \leq T\} \quad (1)$$

Equivalently, each operation bid $B_{ij;a,b}$ is a subset of machine m_{ij} 's object set:

$$O_{m_{ij}} = \{(m_{ij}, t) : 1 \leq t \leq T\} \quad (2)$$

Since preemption of operations is not allowed, the operation bid is restricted as follows:

$$B_{ij;a,b} = \{(m_{ij}, t) : 1 \leq a \leq t \leq b \leq T, b = a + p_{ij} - 1\} \quad (3)$$

Thus, job i 's overall bid is a limited combination of allowed operation bids:

$$B_i = \bigcup_{j:a_{i,j+1} > b_{i,j}} B_{ij;a_{i,j},b_{i,j}} \quad (4)$$

The condition in the set definition determines the precedence constraints between consecutive operations. We will denote the set of all allowed locally feasible bids (i.e., those satisfying non-preemption and precedence constraints) as \mathcal{B}_i .

A job incurs a tardiness cost if its completion is past due, and the job pays a fee for using the resources at its demanded time slots. We thus define job i 's utility function as follows:

$$U_i(B_i) = -W_i \max \{0, b_{i,n_i} - d_i\} - P_i(B_i) \quad (5)$$

where $P_i(B_i)$ is the total payment if the demanded time slots in bid B_i are allocated to job i . First term in the utility function accounts for the total weighted tardiness attributed to job i by demanding B_i which in turn yields the completion of the job at b_{i,n_i} . The total payment P_i is a function of the time slots demanded in bid B_i . Hence, each job agent must trade off possible savings on due-date performance with payments due to resource usage. The best bid for job i (B_i^*) is one that maximizes the utility function, i.e.,

$$\max_{B_i \in \mathcal{B}_i} U_i(B_i). \quad (6)$$

The center has several options in regard to the payment function P_i . A most straight forward payment scheme is *regular tâtonnement* where all job agents are subject to the same linear payment function. There is no price discrimination among job agents and they are charged the same price for a particular (machine, time slot) pair. However, as will become apparent in the following section, under *regular tâtonnement* there is no “optimal pricing” that supports optimal resource allocation. As demonstrated in the context of large-scale linear programming decomposition [22, 23], linear subproblem objective leads to degeneracy. To relieve the problems of degeneracy, Jose, Harker and Ungar [23] propose *augmented tâtonnement* which imposes price discrimination among job agents using a quadratic perturbation term [22] in the subproblem objective. As a result, the unit price paid by a specific job agent varies depending on the particular combination of resource usage it demands. Based on this distinction in the payment (P_i), thus the utility function, we define two types of tâtonnement: *regular*, and *augmented tâtonnement*.

During the progress of the auction, each job agent i solves its locally constrained utility maximization problem to find the best combination of resource-time slots (B_i^*). All job agents then submit their optimal bids to the auctioneer, who collects the new bids, computes and announces the updated resource prices, then proceeds with the next iteration. The optimal bids are $B_i^* \equiv B_i^*(\lambda)$ where $\lambda = (\lambda_1, \dots, M)$, computed from current machine-time slot prices $\lambda_{kt}, k = 1, \dots, M, t = 1, \dots, T$ announced by the auctioneer.

The goal of price updates is to reduce *resource conflicts* among job agents. A conflict exists if one time slot (object) is demanded by more than one job. Since each bid B_i defines a demand point for job i , one way of updating prices in this procedure is to adjust them according to the *excess job demand* (number of jobs that bid for a certain time slot minus the total capacity of the machine). One possible strategy is to have the auctioneer raise the prices in proportion to the excess demand as follows:

$$D_{kt} = \sum_{i=1}^N \delta_{ikt}^* - 1 \quad (7)$$

where δ_{ikt}^* is 1 if job i demands time slot (k, t) in its optimal bid (i.e. if $(k, t) \in B_{ij; a_{ij}, b_{ij}}^*$ and $m_{ij} = k$), is 0 otherwise. Since excess demand can be negative, it is possible to reduce prices for time slots which do not have enough demand. Since no object can be sold with a negative price, we consider only nonnegative prices. Thus we define the price adjustment as follows:

$$\lambda_{kt}^{r+1} = \max \{0, \lambda_{kt}^r + f(D_{kt}^r)\} \quad (8)$$

where r is the iteration number, and f is the *price adjustment function* nondecreasing in current excess demand D_{kt}^r . Since a large variety of functions satisfy this property, this form of price adjustment is quite flexible. Based on the form of f we can define different *auction protocols* that govern the progress from one iteration to the next.

One such auction protocol is called *standard Walrasian tâtonnement* in which adjustment function is defined as a constant multiplier times the current excess demand:

$$f(D_{kt}^r) = s D_{kt}^r \quad (9)$$

where s is called *price adjustment factor* or *step parameter*. This form of tâtonnement with a carefully selected step is known to converge to an optimal (equilibrium) allocation in a pure exchange economy. However, when the demand is discontinuous with interdependencies, one can no longer expect the convergence property to hold in general. Since the step parameter is constant over iterations we call this a *non-adaptive tâtonnement*. The term is defined in contrast to the *adaptive tâtonnement* where the auctioneer makes aggressive price update in early iterations of the auction to quickly assess the overall demand status among job agents. This is followed by smaller adjustments (low values of s) in later iterations to fine tune the quality of allocation. In this case, price adjustment changes depending on the iterations and the progress of the auction.

With the above discussion, we characterize the combinatorial auction by the following elements:

1. Auction Protocols

- *Standard Walrasian tâtonnement* (Non-Adaptive)
- *Adaptive tâtonnement*

2. Payment Functions

- *Regular tâtonnement* (no price discrimination)
- *Augmented tâtonnement* (with price discrimination)

In the next section, we briefly describe a game-theoretic model of this auction-based allocation scheme.

3.3 A Game Theoretic Perspective

We provide a game theoretic interpretation of the above auction model based on the relationship between competitive (or Walrasian) economy and game theory. The model uses the notion of *noncooperative pseudogame* or *abstract economy* proposed by Friedman [15] and Ichiishi [21]. In this game, a player i is influenced by other player in two ways: (1) indirect influence, where i 's feasible strategies is restricted to a subset, and (2) direct influence, where i 's utility level is affected. Consider $N + 1$ players, where players 1 to N are the job agents, and player 0 is the auctioneer. We assume that each job agent has enough endowment to buy any collection of time slots (i.e., no budget restrictions). Job agent i 's consumption set (allowed bids) is also its strategy set, i.e., $S_i = \mathcal{B}_i$. The auctioneer chooses a nonnegative price vector from a strategy set defined as

$$S_0 = \{\lambda_{kt} \geq 0, k = 1, \dots, M, t = 1, \dots, T\}.$$

Given a pair (B, λ) , we define restricted (or feasible) strategy set of player i ($i = 1, \dots, N$) as

$$F_i(B, \lambda) = \{B'_i : \delta'_{ikt} \leq 1 - \sum_{i' \neq i} \delta'_{i'kt}, \forall k, t\}$$

where δ'_i is defined as before according to B'_i . The feasible strategy set of player 0 is not affected by other players, i.e. $F_0 = S_0$. The utility function of player i is defined as

$$u_i((B, \lambda), B'_i) = U_i(B'_i), \forall B'_i \in F_i(B, \lambda)$$

Here, $u_i((B, \lambda), B'_i)$ is agent i 's utility when it chooses B'_i and all others choose

$$(B_1, \dots, B_{i-1}, B_{i+1}, \dots, B_N, \lambda)$$

. The utility function of player 0 is

$$u_n((B, \lambda), \lambda') = \sum_k \sum_t \lambda'_{kt} (\sum_i \delta_{ikt} - 1)$$

One interpretation of this utility maximization is that the auctioneer seeks to minimize the total value of the excess capacity of the machines: $\sum_k \sum_t \lambda_{kt} (-D_{kt})$. Then, the triplet $(\{S_i\}_{i=0}^N, \{F_i\}_{i=0}^N, \{u_i\}_{i=0}^N)$ defines a pseudogame.

It is easy to show that the game corresponding to a classical exchange economy has a strategy equilibrium and that this is a competitive equilibrium. However, this is only true when the utility functions are concave and the strategy sets are convex. Unfortunately these conditions are rarely true in general scheduling problems (see [55] for the existence of equilibrium pricing in one-machine scheduling problems). Nonetheless, the game theoretic perspective introduces potentially interesting issues for distributed scheduling.

3.4 Lagrangean Relaxation

We will now turn our attention to an integer programming (IP) formulation of the job shop scheduling problem with weighted tardiness objective. Although there are several IP formulations for JSP we will use the discrete time formulation due to Pritsker, Watters, and Wolfe [39]. The decision variable is X_{ijt} , where

$$X_{ijt} = \begin{cases} 1 & \text{if operation } j \text{ of job } i \text{ has started by time } t, \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

The JSP under consideration can be formulated as follows:

(JSP)

$$\min \sum_i W_i \left[\sum_{t > d_i - p_{i,n_i}} (1 - X_{i,n_i,t}) \right] \quad (11)$$

$$\text{s.t. } X_{i,j,t+1} \geq X_{i,j,t}, \quad \forall i, j, t < T \quad (12)$$

$$X_{i,j,t} \leq X_{i,j-1,t-p_{i,j-1}}, \quad \forall i, j > 1, t \quad (13)$$

$$\sum_{i,j:m_{ij}=k} (X_{i,j,t} - X_{i,j,t-p_{ij}}) \leq 1, \quad \forall k, t \quad (14)$$

$$X_{i,j,t} \in \{0, 1\}, \quad \forall i, j, t \quad (15)$$

The first set of constraints (12) makes sure that once an operation is started, it remains so in all subsequent time periods. This is due to the definition of X_{ijt} as outlined in (10) and the non-preemption requirement. Constraints (13) state that an operation cannot start until all its predecessors are completed. Finally, the machine capacity constraints (14) state that at most one job can be processed on a particular machine in a given time period. The objective function (11) is the total weighted tardiness derived from the fact that a cost of W_i is incurred for each time period after d_i during which job i has not been completed.

The objective function can be rewritten as follows:

$$\sum_i W_i \left[\sum_{t > d_i - p_{i,n_i}} (1 - X_{i,n_i,t}) \right] = \sum_i W_i (T - d_i + p_{i,n_i}) + \sum_i \sum_j \sum_t -w_{ijt} X_{ijt} \quad (16)$$

where

$$w_{ijt} = \begin{cases} W_i & \text{if } j = n_i \text{ and } t > d_i, \\ 0 & \text{otherwise.} \end{cases} \quad (17)$$

Since the first term in the objective function (16) is a constant we represent it as $A = \sum_i A_i$ where $A_i = W_i (T - d_i + p_{i,n_i})$.

When we dualize the machine capacity constraints (14) of (JSP) with nonnegative Lagrangean multipliers $\lambda_{kt}, \forall k, t$ we obtain the following Lagrangean Relaxation problem:

$$\begin{aligned} (LR_\lambda) \quad & \min \sum_i A_i + \sum_i \sum_j \sum_t (\lambda_{m_{ij},t} - \lambda_{m_{ij},t+p_{ij}} - w_{ijt}) X_{ijt} - \sum_k \sum_t \lambda_{kt} \\ & \text{subject to (12), (13), and (15)} \end{aligned} \quad (18)$$

We decompose LR_λ into independent job-level subproblems as follows:

$$v(LR_\lambda) = \sum_i v(LR_{\lambda,i}) - \sum_k \sum_t \lambda_{kt} \quad (19)$$

where $v(P)$ denotes the value of optimal solution of problem P , and

$$v(LR_{\lambda,i}) = \min A_i + \sum_j \sum_t (\lambda_{m_{ij},t} - \lambda_{m_{ij},t+p_{ij}} - w_{ijt}) X_{ijt} \quad (20)$$

$$\text{s.t. } X_{i,j,t+1} \geq X_{i,j,t} \quad \forall j, t < T \quad (21)$$

$$X_{i,j,t} \leq X_{i,j-1,t-p_{i,j-1}} \quad \forall j > 1, t \quad (22)$$

$$X_{i,j,t} \in \{0, 1\} \quad \forall j, t \quad (23)$$

Each subproblem satisfies the integrality property, in fact each is the dual of a specially structured maximum flow network problem [43]. Hence, each job-level subproblem can be solved using efficient network flow algorithms.

For a vector of Lagrangean multipliers λ , $v(LR_\lambda)$ which is a function of solutions to $(LR_{\lambda,i})$, provides a lower bound on the optimal cost of (JSP) . The best lower bound corresponds to the solution of the following Lagrangean dual problem:

$$(LRD) \quad \max_{\lambda \geq 0} v(LR_\lambda)$$

There are alternative ways to improve the Lagrangean lower bound and solve the Lagrangean dual, among them is the subgradient search procedure [35]. Subgradient optimization starts with an initial value for the multipliers λ , e.g. $\lambda^0 = 0$. Then the method generates a sequence λ^r over the iterations r by the rule

$$\lambda_{kt}^{r+1} = \max\{0, \lambda_{kt}^r + s_r \gamma_{kt}(X^{*r})\} \quad (24)$$

where s_r is an appropriately selected step size and $\gamma_{kt}(X^{*r})$ is the subgradient of the capacity constraint of machine k at time period t defined by the optimal solution X^{*r} of LR_{λ^r} :

$$\gamma_{kt}(X^{*r}) = \sum_{i,j:m_{ij}=k} (X_{i,j,t}^{*r} - X_{i,j,t-p_{ij}}^{*r}) - 1 \quad (25)$$

Step size s_r should satisfy certain conditions in order to get a good convergence to optimal Lagrangean dual value. A formula that has proven effective in practice is as follows [14]:

$$s_r = \alpha_r \left(\frac{UB - v(LR_{\lambda^r})}{\sum_k \sum_t \gamma_{kt}^2(X^r)} \right) \quad (26)$$

where α_r is a scalar satisfying $0 < \alpha_r \leq 2$ and UB is a target upper bound value for Lagrangean dual which can be updated over the iterations.

We may update the upper bound value using a fast heuristic that converts the capacity-infeasible solution (due to resource conflicts) to a feasible one in each iteration. It is also suggested in the literature that one should decrease the value of α if there is no improvement over a certain number of iterations [14]. The procedure stops when (1) a capacity-feasible global solution is found for problem LR_{λ^r} , (2) lower bound is epsilon-close to the upper bound, (3) α becomes very small, or (4) a specified number of iterations is reached. Typically the procedure end with an infeasible solution due to the last three criteria.

3.5 Combinatorial Auction and Lagrangean Relaxation

In this section, we show that subgradient search in the context of Lagrangean Relaxation can be viewed as a particular version of combinatorial auction introduced in Section 3.2. We focus our attention on the payment functions and the auction protocols.

3.5.1 Payment Functions

We first establish a basic property connecting Lagrangean Relaxation with the *regular tâtonnement* version of combinatorial auction. Using the well-known properties of Lagrangean duality we can then explore alternative payment functions.

Theorem 1 *For a given resource price vector λ , let the payment function P_i be the sum of the current resource prices for the slots included in a bid B_i (as in regular tâtonnement), i.e.,*

$$P_i(B_i) = \sum_{j=1}^{n_i} \sum_{t=a_{ij}}^{b_{ij}} \lambda_{m_{ij},t} \quad (27)$$

Then, the following solution $X_{ijt} \forall j, t$ defined using the optimal bid $B_i^(\lambda)$ of job agent i solves the job-level network flow problem $LR_{\lambda,i}$ defined by (20)–(23).*

$$X_{ijt} = \begin{cases} 1 & \text{if } t \geq a_{ij}^* \\ 0 & \text{otherwise.} \end{cases} \quad (28)$$

Proof. First, note that the constraints (21)–(23) are the same constraints that define the set \mathcal{B}_i of allowed bids B_i . If we substitute the defined payment function (27) into the utility function $U_i(B_i)$ (see Equation (5)), we obtain

$$U_i(B_i) = -W_i \max \{0, b_{i,n_i} - d_i\} - \sum_{j=1}^{n_i} \sum_{t=a_{ij}}^{b_{ij}} \lambda_{m_{ij},t}$$

We note that, for a given λ , this is negative of the objective (20) of the job-level problem $(LR_{\lambda,i})$. Then, the maximum value of the constrained utility function is the same as the minimum value of the job-level objective in LR. Since $B_i^*(\lambda)$ is optimal, it solves the utility maximization problem defined by (6). Hence,

$$U_i(B_i^*) = -W_i \max \{0, b_{i,n_i}^* - d_i\} - \sum_{j=1}^{n_i} \sum_{t=a_{ij}^*}^{b_{ij}^*} \lambda_{m_{ij},t} = v(LR_{\lambda,i})$$

where a_{ij}^* and b_{ij}^* respectively represent the first and last time slots for each operation j included in the bid B_i^* . That is, a_{ij}^* defines the optimal starting time of the operation j of job i in optimal bid B_i^* . Hence, can construct the solution (28) using the definition of decision variables X .

Theorem 2 *Let λ^* be the optimal resource prices corresponding to the Lagrangean Dual Problem (LRD), i.e.,*

$$\lambda^* = \arg \max_{\lambda \geq 0} v(LR_{\lambda}),$$

and $B_i^(\lambda^*)$ be the optimal bids for price vector λ^* where*

$$U_i(B_i^*) = v(LR_{\lambda^*}).$$

Locally optimal bids $B_i^(\lambda^*)$ solves (JSP) if and only if λ^* and B_i^* satisfy the Strong Lagrangean Duality Theorem [35].*

Proof. This follows directly from Theorem 1 and the well-known Strong Lagrangean Duality Theorem.

Only in rare cases do discrete optimization problems satisfy the requirement of Strong Lagrangean Duality. In the case of (JSP), a significant duality gap between $v(LR_{\lambda^*})$ and $v(JSP)$ is to be expected. This result implies that there is unlikely to be optimal resource

prices (therefore bids) that would support the optimal solution to JSP. In other words, if linear resource pricing with no price discrimination is used in combinatorial auction (as is the case in *regular tâtonnement*), the auction procedure does not lead to an optimal resource allocation. A market clearing mechanism (in this case, a feasibility restoration heuristic, or a branch and bound algorithm) will be necessary at the end of the auction to resolve the resource conflicts. As discussed in Section 3.2, even in the case of a large scale linear program (which does satisfy Strong Lagrangean Duality at optimality), there may not be resource prices supporting the optimal solution due to subproblem degeneracy.

The above results motivate the consideration of price discrimination in the job payment function, which leads to an *augmented tâtonnement* [22, 23]. In this case, the center announces a *price schedule* instead of a price vector:

$$\lambda'_{kt} = \lambda_{kt} + q\delta_{ikt} \quad (29)$$

where λ_{kt} is the original price, q is a carefully selected small constant for better convergence, and δ_{ikt} is defined as before (i.e., 1 if time slot t on machine k is demanded by job i , 0 otherwise). Thus, job i 's payment function is defined as:

$$P_i(B_i) = \sum_{j=1}^{n_i} \sum_{t=a_{ij}}^{b_{ij}} \lambda'_{m_{ij},t} \quad (30)$$

This change is reflected in the objective of the i 'th subproblem in LR as follows:

$$A_i + \sum_j \sum_t -w_{ijt}X_{ijt} + \sum_j \sum_t \lambda_{m_{ij},t}(X_{ijt} - X_{i,j,t-p_{ij}}) + \sum_j \sum_t q(X_{ijt} - X_{i,j,t-p_{ij}})^2 \quad (31)$$

However, the standard price schedule (29) applied to (*JSP*) has an anomaly as the quadratic perturbation term added to the objective joins to the linear term (with the multiplier q) since

$$(X_{ijt} - X_{i,j,t-p_{ij}})^2 = X_{ijt} - X_{i,j,t-p_{ij}}.$$

To address this issue, we define a different perturbation scheme which adds a quadratic term for each *time zone* defined as a combination of fixed number of consecutive time slots. The revised objective is as follows:

$$A_i + \sum_j \sum_t -w_{ijt}X_{ijt} + \sum_j \sum_t \lambda_{m_{ij},t}(X_{ijt} - X_{i,j,t-p_{ij}}) + \sum_j \sum_{h=1}^H q \left(\sum_{t=T_h}^{T_h+L-1} X_{ijt} - X_{i,j,t-p_{ij}} \right)^2 \quad (32)$$

where H is the number of time zones each with length L and beginning time of a zone h is defined by T_h . We can then define the time zone capacity as the sum of L consecutive

capacity constraints in (14):

$$\sum_{t=T_h}^{T_h+L-1} \sum_{i,j:m_{ij}=k} (X_{i,j,t} - X_{i,j,t-p_{ij}}) \leq L, \quad \forall k, t \quad (33)$$

We now have two alternative utility functions defined based on the specific payment function set by the center:

- *Regular tâtonnement* where payment function P_i is defined as in Equation (27).
- *Augmented tâtonnement* in which payment function P_i includes a quadratic perturbation defined in Equation (32).

Note that for the latter case each job agent needs to solve a quadratic subproblem. This implies a higher computational cost per iteration, but the overall convergence might compensate for this extra cost.

3.5.2 Auction Protocols

As summarized earlier, auction protocols can be implemented in a non-adaptive *standard Walrasian* fashion, or as an *adaptive tâtonnement*. The subgradient optimization (Section 3.4) can be viewed as a particular version of the latter. While the subgradient optimization tries to penalize infeasibility of dualized constraints, the auctioneer updates the resource prices so as to discourage conflicts on the demanded objects. While the subgradient algorithm adjusts the prices proportional to the amount of infeasibility, the auction algorithm updates the prices proportional to desirability of the object (measured in terms of excess demand). In fact, the subgradient (25) defined for the optimization procedure can be set equal to the excess demand (7) used in the auction:

$$\gamma_{kt}(X^{*r}) = D_{kt}^r. \quad (34)$$

Then, one adaptive way of updating prices is to set the adjustment function f in (8) as below:

$$f(D_{kt}^r) = s_r D_{kt}^r \quad (35)$$

In this case, the auction algorithm can be set equivalent to the subgradient optimization. Moreover, the basic principle of price adjustment in an *adaptive tâtonnement* is parallel to the idea of subgradient price adjustments. The proven convergence properties of subgradient search are particularly attractive. Nonetheless, at the end of the subgradient optimization a significant duality gap is to be expected and a feasibility restoration heuristic must be used to eliminate the positive excess demand.

We now define in more specific terms the two auction protocols based on the definition of the price adjustment function (8):

- *Non-adaptive tâtonnement*, $f(D_{kt}^r) = sD_{kt}^r$.
- *Adaptive tâtonnement*, $f(D_{kt}^r) = s_r D_{kt}^r$ where s_r is the step size defined in (26).

The combinatorial auction mechanism for scheduling is summarized in Figure 1. As shown in the procedure, the auctioneer is not only responsible for price update in each iteration, it also performs feasibility restoration in each iteration. In this way, the auctioneer may announce the best potential feasible schedule in case the auction terminates early. This can be viewed as an upper bound calculation routine used in each iteration. Note that each job agent's local problem is a maximum flow problem, for which there exist polynomial algorithms.

4 Examples

Combining the two different payment functions and the two auction protocols leads to four different versions of combinatorial auction. In the following, we use small job shop examples to illustrate the implementation details of the proposed auction algorithm. We are also interested in gaining insights on the convergence behavior as well as the quality (duality gap) of the final solution when the auction terminates.

We first study small (3 machines, 3 jobs, or 3x3) scheduling problems. Since the routing structure of the jobs dictates both the level and the sequence in which the jobs demand for resources, we expect the auction mechanism to behave quite differently under different

- Step 1. Initialization: The auctioneer initializes the machine-time slot prices λ , ($\lambda_{kt}^0 = 0$, $\forall k, t$); set iteration counter $r = 0$, scalar parameter α_r ($\alpha_0 = 2$), and the multiplier of the quadratic term ($q = 0.1$) (for augmented tâtonnement only).
- Step 2. Each job agent solves the job-level utility maximization subproblem (using either one of the payment functions) then submits the its optimal bid ($B_i^*(\lambda^r)$) to the auctioneer. Each bid corresponds to a job-level schedule.
- Step 3. The auctioneer combines all the bids and generates a capacity-infeasible shop-level schedule. The summation of total weighted tardiness and total potential payments from the jobs gives a lower bound (LB_r) for the original JSP.
- Step 4. The auctioneer converts this capacity-infeasible schedule into a feasible one by resolving the resource conflicts. Using this feasibility-restored shop schedule, the auctioneer computes corresponding total weighted tardiness as an upper bound (UB_r).
- Step 5. The auctioneer updates the best upper bound value found so far, $UB \leftarrow UB_r$, and records the best feasible shop schedule. The auctioneer updates α_r if necessary (Halve it if there is no improvement in LB_r in last 3 iterations).
- Step 6. The auctioneer computes the excess demand vector (D), (step size s_r in the case of *adaptive tâtonnement*) and updates time-slot prices λ^{r+1} using one of the protocols.
- Step 7. Checks if a stopping criterion is satisfied. If not, the auctioneer starts the next iteration $r \leftarrow r + 1$, go to Step 2. Otherwise, stop, and announce the best feasible schedule found.

Figure 1: An iterative combinatorial auction mechanism for resource scheduling

routing structures. We choose three different routing structures for our examples: (1) *random job shop*: the sequence each job visits the machines is random, (2) *bottleneck job shop*: the routing is random as in (1), but all jobs require longer processing times on a specific bottleneck machine, and (3) *flow shop*: the jobs visit the machines in the same order, but require different processing times. We investigate each of the three routing types in the following sections.

4.1 Random job shop

The data for this problem is shown in Table 2 (This is a modified version of the 3x3 job shop example in [24]). First, we implement the *adaptive regular tâtonnement* in which the payment function has no price discrimination and the auction protocol is adaptive as

Table 2: 3x3 Random job shop example

Job	Weight	Due date	Machine (Processing time)
1	4	10	1 (3), 2 (1), 3 (6)
2	6	10	3 (3), 1 (7), 2 (1)
3	2	12	1 (2), 3 (4), 2 (4)

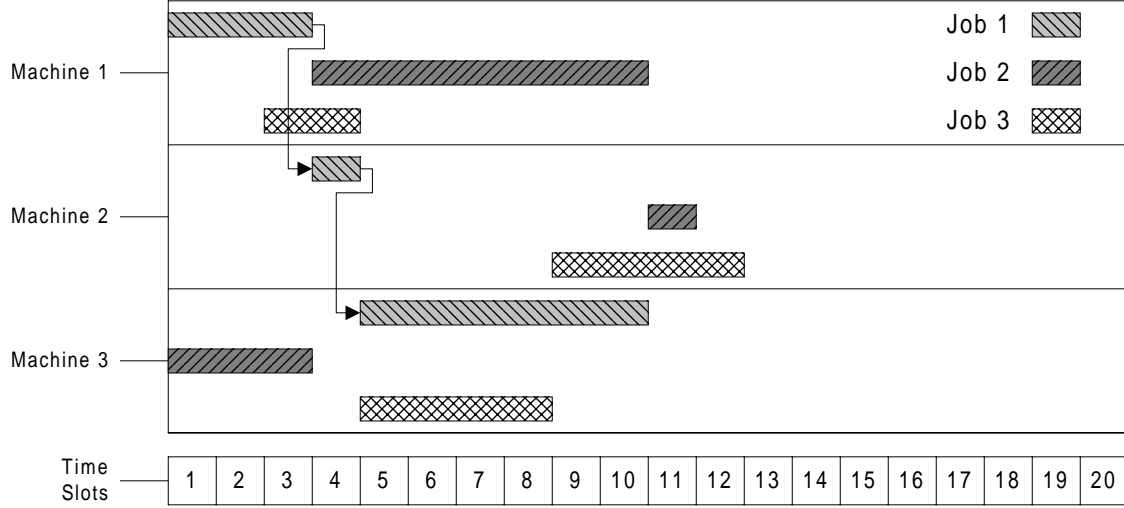


Figure 2: Infeasible shop schedule after combining the job bids (first iteration)

in subgradient search. We set the planning horizon (T) as 30 time periods. Hence, we have 30 time slots (objects) in each machine and 90 objects in total that the job agents must bid for. The capacity-infeasible shop schedule constructed by the auctioneer after the first iteration is shown in Figure 2. As shown in the figure, the job-level schedules satisfy non-preemption and precedence constraints, the machine capacity constraints are violated in a few time slots. The auctioneer will increase the prices of these time slots in the following iteration and thereby reducing the demand. The feasibility restored schedule after iteration 4 is shown in Figure 4.

For the examples problems in this section, we restricted the number of iterations (K) to 30. The price profile for each machine at the end of the auction is shown in Figure 3. As we can see, each machine has a considerably different price profile over the time slots. The adaptive subgradient price update essentially captures the desirability of each time

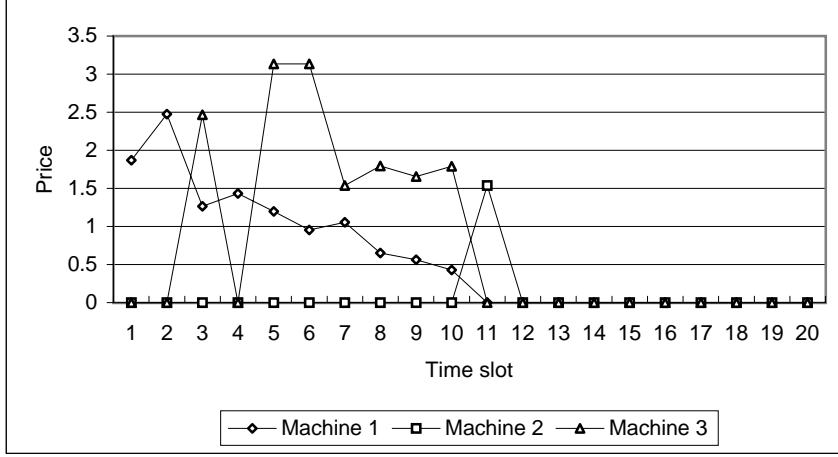


Figure 3: Price profiles at the end of *adaptive regular tâtonnement*

slot on each machine. At the end of this adaptive tâtonnement, the auctioneer announce the best overall schedule generated over the iterations. Figure 5 plots LB and the best UB values versus iterations. Major improvements on LB are in early iterations. The final duality gap for this problem is 14.6%. For this example problem, the best UB schedule with the total weighted tardiness of 22 (achieved in iteration 4) is optimal (The optimality proof is done outside the algorithm by solving the problem monolithically; it is shown in Figure 4).

The iterative process in *Non-Adaptive tâtonnement* is similar to the adaptive version except that the step parameter (s) is constant over the iterations. We experimented with several s values but we will only summarize the results for the case of $s = 0.2$. The price profile achieved at the end is similar to that shown in Figure 3. The change in LB and UB over iterations is plotted in Figure 6. While LB improves slower than in the adaptive process, the best UB is achieved in a later iteration (9th).

We next study *augmented tâtonnement* with non-adaptive and adaptive price updates. Recall that the augmented approach requires the use of “time zones.” We constructed each time zone by combining two consecutive time slots, i.e., $L = 2$ and $H = 15$. In *augmented tâtonnements*, the selection of the parameter q directly affects the convergence behavior. For the example problems, we experimented with several values and fixed q at 0.1. For the *adaptive augmented tâtonnement*. Note that the LB and UB are now computed by solving job agent subproblems with a revised quadratic objective function.

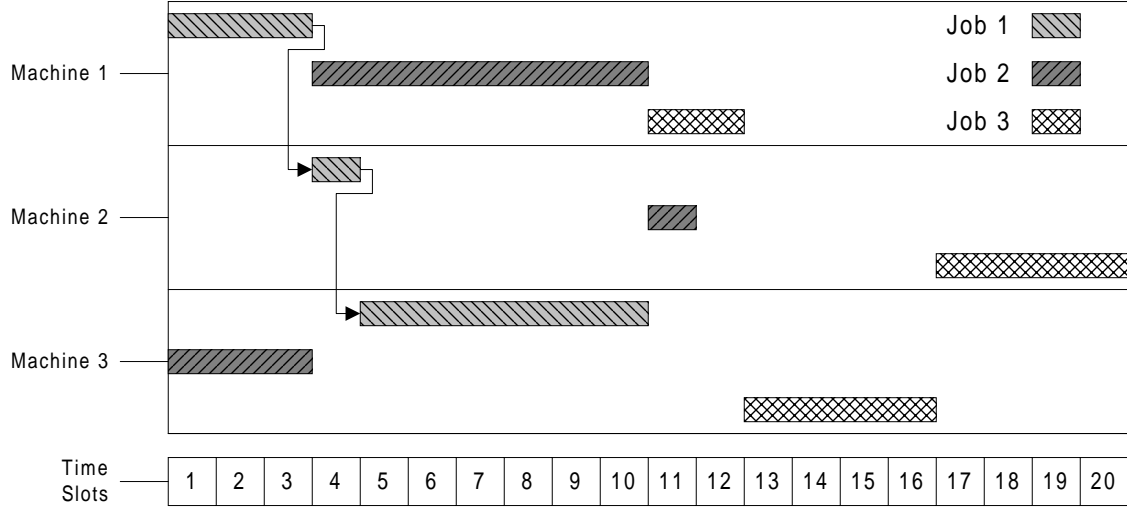


Figure 4: Feasibility-restored shop schedule after the fourth iteration (the best UB schedule which is also optimal).

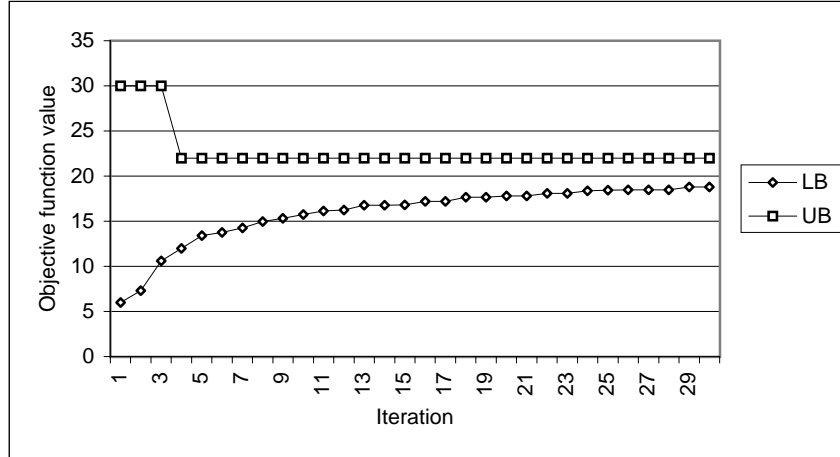


Figure 5: Lower bound (LB) and Upper bound (UB) during the *adaptive regular tâtonnement*

The revised LB and UB values are shown in Figure 7 with a final duality gap of 23.4%. The best restored feasible schedule (in fact, it is the optimal schedule shown in Figure 4) is found in iteration 3.

The implementation for *non-adaptive augmented tâtonnement* is straight-forward. Al-

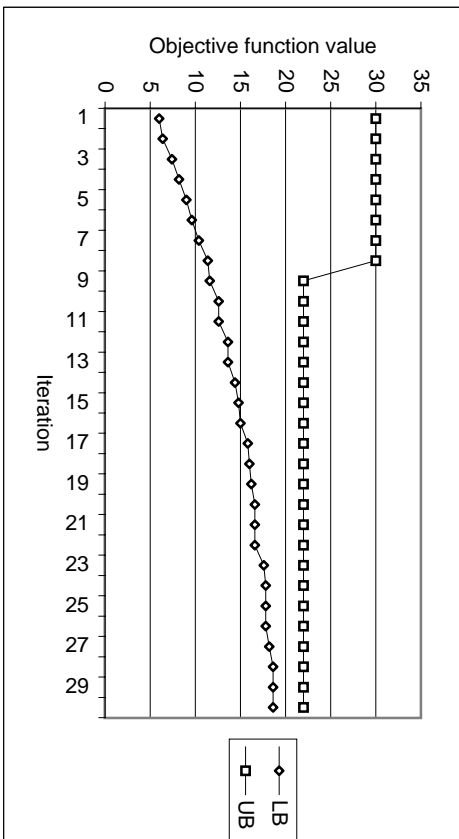


Figure 6: Lower bound (LB) and Upper bound (UB) values during the *non-adaptive regular tâtonnement*

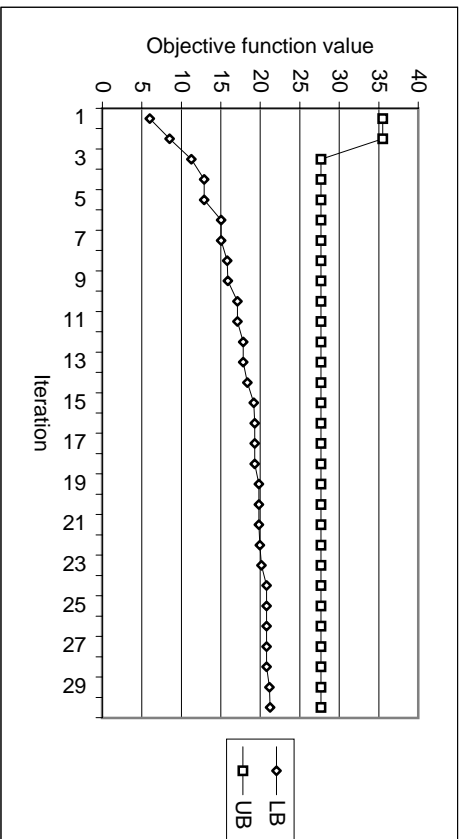


Figure 7: Lower bound (LB) and Upper bound (UB) values during the *adaptive augmented tâtonnement*

though there is no need to calculate lower and upper bound values during the *tâtonnement*, we keep track of the values to observe its convergence behavior and the quality of the final allocation (See Figure 8). The best feasible schedule is reached in iteration 7, and somewhat larger duality gap (32%) results at the end as compared to adaptive version.

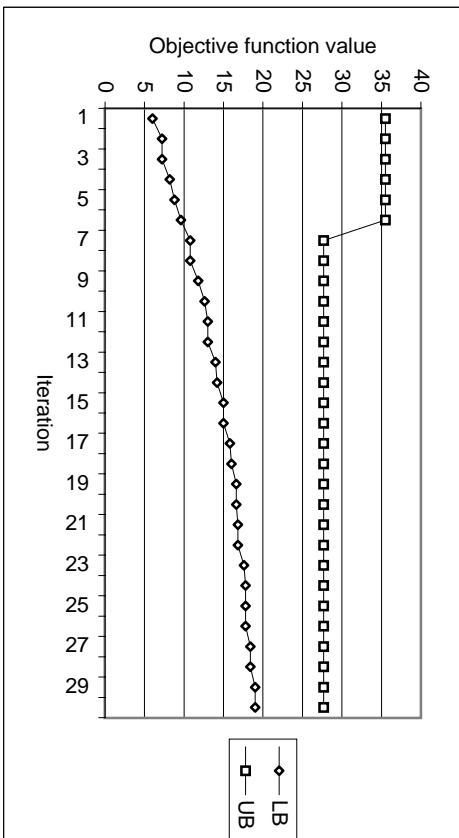


Figure 8: Lower bound (LB) and Upper bound (UB) during the *non-adaptive augmented tâtonnement*

Table 3: 3x3 Bottleneck job shop example

Job	Weight	Due date	Machine (Processing time)
1	3	12	1 (7), 2 (3), 3 (2)
2	6	11	3 (3), 1 (6), 2 (2)
3	4	9	3 (1), 1 (5), 2 (3)

4.2 Bottleneck Job Shop

This example is similar to the random job shop case except that there is a bottleneck machine (machine 1) where all jobs require a longer processing time (See Table 3). Figure 9 shows the price profile at the end of *adaptive regular tâtonnement*. As the figure shows, almost all time slots, except the ones on the bottleneck machine, remain at their initial prices of zero. This should come with little surprise since the bottleneck machine is subject to the most competition among the job agents. As Figure 10 shows that UB and LB improve significantly over the iterations reaching a final duality gap of only 2.6%. The UB schedule achieved in 7th iteration is optimal with weighted tardiness value of 54.

The implementation of *non-adaptive regular tâtonnement* yields similar results as summarized in Figure 11. The best feasible solution is found in iteration 10.

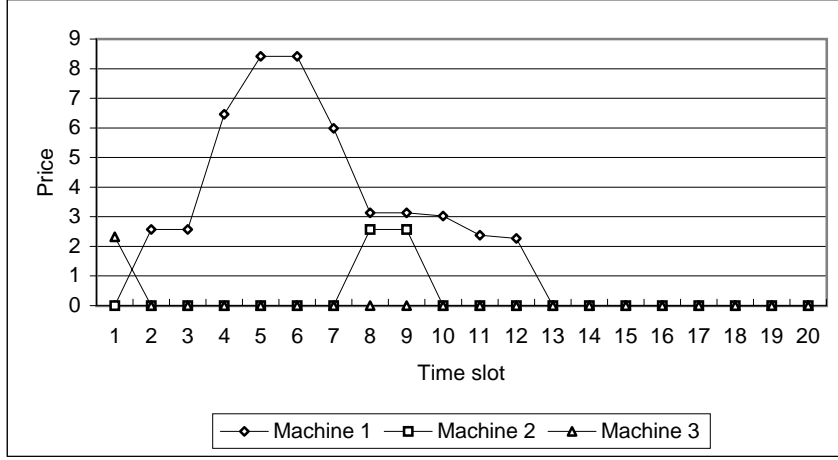


Figure 9: Price profiles for each machine at the end of *adaptive regular tâtonnement* (Bottleneck Job Shop)

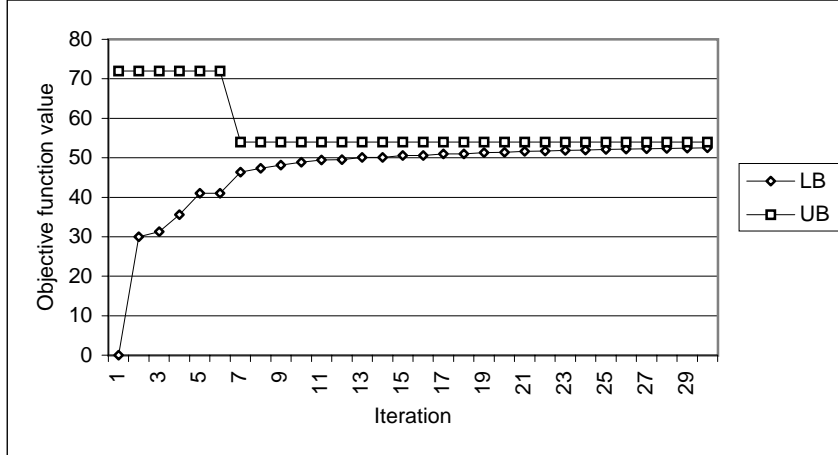


Figure 10: Lower bound (LB) and Upper bound (UB) values during the *adaptive regular tâtonnement* (Bottleneck Job Shop)

The *augmented tâtonnement* with the adaptive auction protocol yields results as shown in Figure 12. Using perturbed payment function helps achieve the best upper bound in the second (2nd) iteration. The results for its non-adaptive counterpart are very similar to the adaptive version and are not presented.

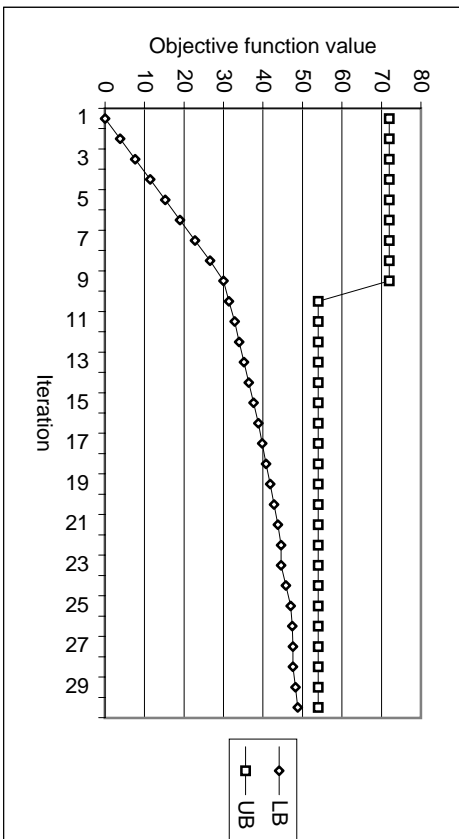


Figure 11: Lower bound (LB) and Upper bound (UB) values during the *non-adaptive regular tâtonnement* (Bottleneck Job Shop)

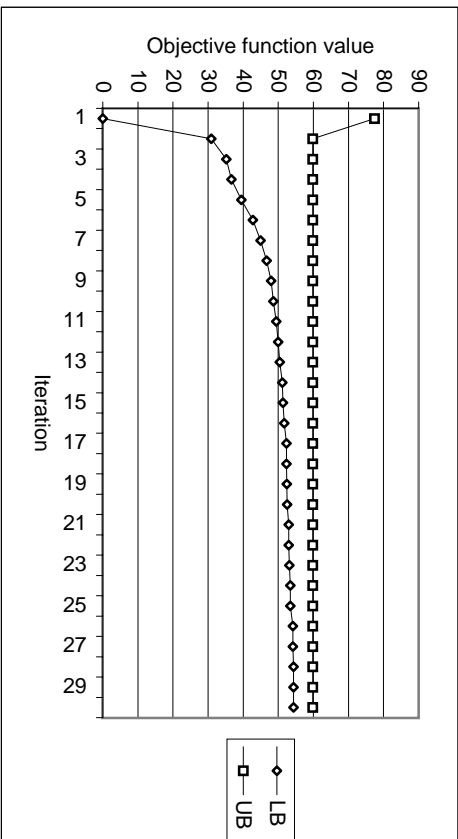


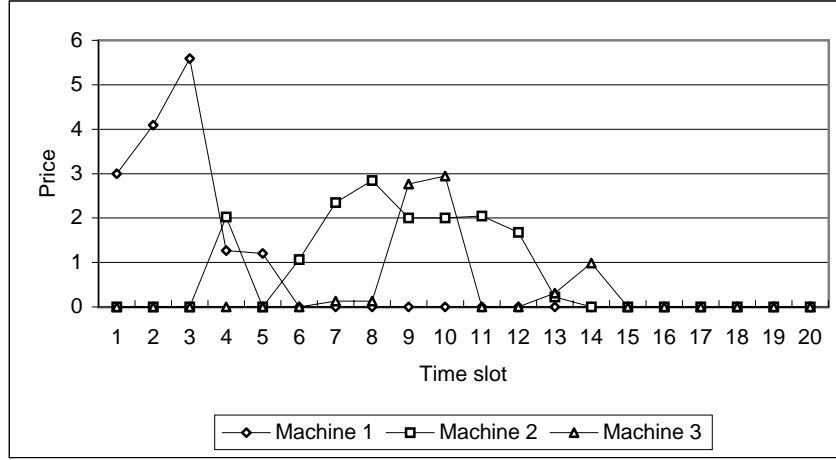
Figure 12: Lower bound (LB) and Upper bound (UB) values in *adaptive augmented tâtonnement* (Bottleneck Job Shop)

4.3 Flow Shop

The flow shop data is shown in Table 4. Note that the jobs visit the machines in the same order but their processing times differ. The price profiles reached at the end of the *adaptive regular tâtonnement* are depicted in Figure 13. As one may expect, the first machine has higher prices on the earlier time slots, while the second and third machines have positive prices in later time periods. As can be seen from Figure 14, the best upper

Table 4: 3x3 Flow shop example

Job	Weight	Due date	Machine (Processing time)
1	4	10	1 (3), 2 (1), 3 (6)
2	6	13	1 (3), 2 (7), 3 (1)
3	2	12	1 (2), 2 (4), 3 (4)

Figure 13: Price profiles at the end of *adaptive regular tâtonnement* (Flow Shop)

bound (which is also optimal) is reached in third iteration, and the final duality gap is 17.2%. If we compare the duality gaps between different routing structures, we see that the bottleneck job shop has the smallest, the flow shop the next, and the job shop the largest.

Revised values of LB and UB for *augmented tâtonnement* with the adaptive auction protocol are depicted in Figure 15. It reaches the optimal solution in second iteration. The results of the non-adaptive version of both the regular and the augmented tâtonnements are not presented for brevity, since they are consistent with the earlier observations (i.e. slower convergence, wider duality gaps).

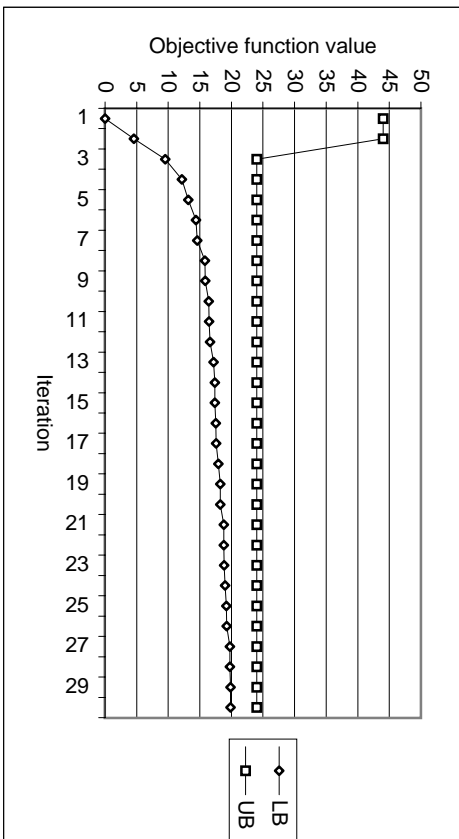


Figure 14: Lower bound (LB) and Upper bound (UB) values during the *adaptive regular tâtonnement* (Flow Shop)

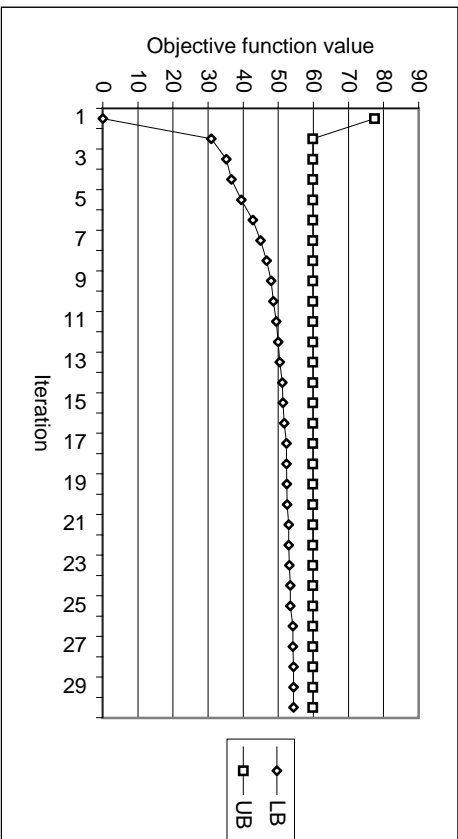


Figure 15: Lower bound (LB) and Upper bound (UB) during *adaptive augmented tâtonnement* (Flow shop)

5 Conclusions

In this paper, we introduced a group of price-directed auction mechanisms for distributed scheduling. We investigated two *auction protocols* (non-adaptive Walrasian and adaptive tâtonnements) and two *payment functions* (regular and augmented tâtonnements). We showed that the well-known Lagrangian Relaxation using subgradient search corresponds to an adaptive regular tâtonnement. We demonstrate the operations of these auction

mechanisms using small examples representing three major routing structures (random job shop, bottleneck job shop, and flow shop). As demonstrated in the examples, the prices of time slots (objects) depend heavily on the demand patterns (i.e., routing structures and processing times) of job agents.

This paper presents a step toward answering many important questions concerning distributed scheduling. By exploring the relationship among auction tâtonnements, discrete optimization algorithms, and distributed decision making, we are able to categorize different alternatives for implementation and their possible implications. The computational examples show that the adaptive price update with subgradient step might be superior to the non-adaptive auction protocol. Moreover, augmented tâtonnement using quadratic price discrimination might be effective in speeding up the convergence. Of course, a much more extensive computational study is necessary to generalize these results. Other generalization may include the cases where each agent responsible for several jobs. In this case, the agents' local problem will be more difficult to solve, but on the other hand, less resource conflicts may result.

In this study, we do not address the issue of incentive compatibility, i.e., how and why the job agents decide to participate in the auction and how to design incentives so that the job agents reveal their true preferences. We assume that all job agents will participate in the auction and will report their true optimal solutions. Earlier studies show that some price-directed mechanisms that guarantee the convergence to a globally optimal solution are incentive compatible in certain cases [8, 9]. There is no research to-date for the more complex discrete optimization problems. We believe that in order to achieve true distributed scheduling in realistic settings, the above issues and many other complicating factors need to be carefully addressed.

Acknowledgment

The research is supported, in part, by National Science Foundation Grant DMI-9634808. We wish to thank Professor Patrick T. Harker at the Wharton School of the University of Pennsylvania for many valuable suggestions during the research.

References

- [1] A. D. Baker. Metaphor or reality: A case study where agents bid with actual costs to schedule a factory. In S. H. Clearwater, editor, *Market-Based Control: A Paradigm for Distributed Resource Allocation*. World Scientific, River Edge, NJ, 1996.
- [2] J. S. Banks, J. O. Ledyard, and D. P. Porter. Allocating uncertain and unresponsive resources. *RAND Journal of Economics*, 20(1):1–25, 1989.
- [3] D. P. Bertsekas. The auction algorithm: A distributed relaxation method for the assignment problem. *Annals of Operations Research*, 14:105–123, 1988.
- [4] D. P. Bertsekas. The auction algorithm for assignment and other network flow problems: A tutorial. *Interfaces*, 20(4):133–149, 1990.
- [5] D. P. Bertsekas. *Linear Network Optimization*. MIT Press, Cambridge, MA, 1991.
- [6] D. P. Bertsekas. Auction algorithms for network flow problems. *Computational Optimization and Applications*, 1:7–66, 1992.
- [7] S. Bikhchandani and J. W. Mamer. Competitive equilibrium in an exchange economy with indivisibilities. *Journal of Economic Theory*, 74:385–413, 1997.
- [8] S. I. Cohen. Incentives, iterative communication, and organizational control. *Journal of Economic Theory*, 22:37–55, 1980.
- [9] S. I. Cohen. Truth-telling, dominant strategies, and iterative Groves mechanisms. *Public Choice*, 51:333–343, 1986.
- [10] R. Davis and R. G. Smith. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, 20:63–109, 1983.
- [11] G. Demange, D. Gale, and M. Sotomayor. Multi-item auctions. *Journal of Political Economy*, 94(4):863–872, 1986.
- [12] R. Engelbrecht-Wiggans. Auctions and bidding models: A survey. *Management Science*, 26(2):119–142, 1980.
- [13] R. Engelbrecht-Wiggans. An introduction to the theory of bidding for a single object. In R. Engelbrecht-Wiggans, M. Shubik, and R. M. Stark, editors, *Auctions, Bidding, and Contracting: Uses and Theory*, pages 53–103. New York University Press, New York, NY, 1983.

- [14] M. L. Fisher. An application oriented guide to Lagrangian relaxation. *Interfaces*, 15(2):10–21, 1985.
- [15] J. W. Friedman. *Game theory with applications to economics*. Oxford University Press, New York, NY, 1990.
- [16] R. L. Graves, L. Schrage, and J. K. Sankaran. An auction method for course registration. *Interfaces*, 23(5):81–92, 1993.
- [17] F. Gul and E. Stacchetti. English and double auctions with differentiated commodities. Technical Report 97-02, University of Michigan, 1997.
- [18] F. Gul and E. Stacchetti. Walrasian equilibrium without complementarities. Technical Report 97-01, University of Michigan, 1997.
- [19] A. Hylland and R. Zeckhauser. An efficient allocation of individuals to positions. *Journal of Political Economy*, 87(2):293–314, 1979.
- [20] T. Ichiishi. *Game theory for Economic Analysis*. Academic Press, New York, NY, 1983.
- [21] P. Jennergren. A price schedules decomposition algorithm for linear programming problems. *Econometrica*, 41(5):965–979, 1973.
- [22] R. A. Jose, P. T. Harker, and L. H. Ungar. Auction and optimization: Methods for closing the gap caused by discontinuities in demands. Technical report, University of Pennsylvania, 1997.
- [23] B. Lageweg, J. K. Lenstra, and A. H. G. Rinnooy Kan. Job shop scheduling by implicit enumeration. *Management Science*, 24:441–450, 1977.
- [24] H. B. Leonard. Elicitation of honest preferences for the assignment of individuals to positions. *Journal of Political Economy*, 91(3):461–479, 1983.
- [25] J. Liu and K. P. Sycara. Distributed constraint satisfaction through constraint partition and coordinated reaction. In *Proceedings of the 12th International Workshop on Distributed Artificial Intelligence*, 1993.
- [26] J. Liu and K. P. Sycara. Distributed problem solving through coordination in a society of agents. In *Proceedings of the 13th International Workshop on Distributed Artificial Intelligence*, 1994.

- [27] J. Liu and K. P. Sycara. Exploiting problem structure for distributed constraint optimization. In *Proceedings of the 1st International Conference on Multi-agent Systems*, San Francisco, CA, 1995.
- [28] J. Ma. English auctions and walrasian equilibria with multiple objects: A dynamic approach. Technical Report 97-02, Rutgers University, 1997.
- [29] J. Ma. Competitive equilibrium with indivisibilities. Technical Report 98-09, Rutgers University, 1998.
- [30] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University Press, New York, NY, 1995.
- [31] R. P. McAfee and J. McMillan. Auctions and bidding. *Journal of Economic Literature*, 25(2):699–738, 1987.
- [32] P. R. Milgrom and R. J. Weber. A theory of auctions and competitive bidding. *Econometrica*, 50(5):1089–1122, 1982.
- [33] D. E. Neiman, D. W. Hildum, V. R. Lesser, and T. W. Sandholm. Exploiting meta-level information in a distributed scheduling system. In *Proceedings of the 12th Conference on Artificial Intelligence*, Seattle, WA, 1994.
- [34] R. G. Parker and R. L. Rardin. *Discrete Optimization*. Academic Press, Inc., San Diego, CA, 1988.
- [35] H. V. D. Parunak. Manufacturing experience with the contract net. In M. N. Huhns, editor, *Distributed Artificial Intelligence*, pages 285–310. Morgan Kaufmann Publishers, Los Altos, CA, 1987.
- [36] H. V. D. Parunak. Applications of distributed artificial intelligence. In G. M. P. O’Hare and N. R. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, pages 139–164. John Wiley and Sons, 1996.
- [37] M. Pinedo. *Scheduling: Theory, Algorithms and Systems*. Prentice Hall, Englewood Cliffs, NJ, 1995.
- [38] A. Pritsker, L. Watters, and P. Wolfe. Multiproject scheduling with limited resources: A zero-one programming approach. *Management Science: Theory*, 16(1):93–108, 1969.

- [39] S. J. Rassenti, V. L. Smith, and R. L. Bulfin. A combinatorial auction mechanism for airport time slot allocation. *Bell Journal of Economics*, 13:402–417, 1982.
- [40] M. H. Rothkopf and R. M. Harstad. Modeling competitive bidding: A critical essay. *Management Science*, 40(3):364–384, 1994.
- [41] M. H. Rothkopf, A. Pekec, and R. M. Harstad. Computationally manageable combinatorial auctions. Technical report, Rutgers University, 1995.
- [42] R. O. Roundy, W. L. Maxwell, Y. T. Herer, S. R. Tayur, and A. W. Getzler. A price-directed approach to real-time scheduling of production operations. *IIE Transactions*, 23(2):149–160, 1991.
- [43] N. Sadeh. MICRO-BOSS: A micro-opportunistic factory scheduler. Technical Report CMU-RI-TR-91-22, Carnegie Mellon University, The Robotics Institute, 1991.
- [44] T. Sandholm. An implementation of contract net protocol based on marginal cost calculations. In *Proceedings of the 11th National Conference on Artificial Intelligence*, pages 256–262, Washington D.C., 1993.
- [45] J. K. Sankaran. On a dynamic auction mechanism for a bilateral assignment problem. *Mathematical Social Sciences*, 28(2):143–150, 1994.
- [46] V. L. Smith. *Papers in Experimental Economics*. Cambridge University Press, New York, NY, 1991.
- [47] R. M. Stark and M. H. Rothkopf. Competitive bidding: A comprehensive bibliography. *Operations Research*, 27:364–391, 1979.
- [48] K. Sycara, S. Roth, N. Sadeh, and M. Fox. Distributed constraint heuristic search. *IEEE Transactions on System, Man and Cybernetics*, 21:1446–1461, 1991.
- [49] D. M. Upton, M. M. Barash, and A. M. Matheson. Architectures and auctions in manufacturing. *International Journal of Computer Integrated Manufacturing*, 4(1):23–33, 1991.
- [50] W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance*, 16(1):8–37, 1961.
- [51] K. Wang and D. Veeramani. An adaptive machine bidding strategy for distributed shop-floor control under stochastic part demands. In *5th Industrial Engineering Research Conference Proceedings*, pages 321–326, Minneapolis, MN, 1995.

- [52] M. P. Wellman. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research*, 1(1):1–23, 1993.
- [53] M. P. Wellman. Market-oriented programming: Some early lessons. In S. H. Clearwater, editor, *Market-Based Control: A Paradigm for Distributed Resource Allocation*. World Scientific, River Edge, NJ, 1996.
- [54] M. P. Wellman, W. E. Walsh, P. R. Wurman, and J. K. MacKie-Mason. Some economics of market-based distributed scheduling. In *Eighteenth International Conference on Distributed Computing Systems*, Amsterdam, The Netherlands, 1998.
- [55] F. Ygge and H. Akkermans. On resource-oriented multi-commodity market computations. In *Proceedings of the 4th International Conference on Multi-Agent Systems ICMAS'98*, 1998.

September 30, 1998